Most General Explanations of Tree Ensembles

Yacine Izza[1] , **Alexey Ignatiev**[2] , **Sasha Rubin**[3] , **Joao Marques-Silva**[4] , **Peter J. Stuckey**[2,5]
[1]CREATE, National University of Singapore, Singapore
[2]Monash University, Melbourne, Australia
[3]University of Sydney, Australia
[4]ICREA, University of Lleida, Spain
[5]OPTIMA ARC Industrial Training and Transformation Centre, Melbourne, Australia
izza@comp.nus.edu.sg, alexey.ignatiev@monash.edu, sasha.rubin@sydney.edu.au, jpms@icrea.cat,
peter.stuckey@monash.edu

Abstract

Explainable Artificial Intelligence (XAI) is critical for attaining trust in the operation of AI systems. A key question of an AI system is "why was this decision made this way". Formal approaches to XAI use a formal model of the AI system to identify *abductive explanations*. While abductive explanations may be applicable to a large number of inputs sharing the same concrete values, more general explanations may be preferred for numeric inputs. So-called *inflated abductive explanations* give intervals for each feature ensuring that any input whose values fall withing these intervals is still guaranteed to make the same prediction. Inflated explanations cover a larger portion of the input space, and hence are deemed more general explanations. But there can be many (inflated) abductive explanations for an instance. Which is the best? In this paper, we show how to find a most general abductive explanation for an AI decision. This explanation covers as much of the input space as possible, while still being a correct formal explanation of the model's behaviour. Given that we only want to give a human one explanation for a decision, the most general explanation gives us the explanation with the broadest applicability, and hence the one most likely to seem sensible.

1 Introduction

The widespread use of artificial intelligence to make or support decisions effecting humans has led to the need for these systems to be able to explain their decisions. The field of eXplainable AI (XAI) has emerged in response to this need. Its generally accepted that XAI is required to deliver trustworthy AI systems. Unfortunately the bulk of work in XAI offers no formal guarantees, or even clear definitions of the "meaning" of an explanation. Examples of non-formal XAI include model-agnostic methods [Ribeiro *et al.*, 2016; Lundberg and Lee, 2017; Ribeiro *et al.*, 2018], heuristic learning of saliency maps (and their variants) [Samek *et al.*, 2019; Samek *et al.*, 2021], but also proposals of intrinsic interpretability [Rudin, 2019; Molnar, 2020; Rudin *et al.*, 2022]. In recent years, comprehensive evidence has been gathered that attests to the lack of rigor of these (non-formal) XAI approaches [Slack *et al.*, 2020; Ignatiev, 2020; Wu *et al.*, 2023; Marques-Silva and Huang, 2024].

Formal XAI [Shih *et al.*, 2018; Ignatiev *et al.*, 2019b; Marques-Silva and Ignatiev, 2022] in contrast provides rigorous definitions of explanations which have desirable properties as well as practical algorithms to compute them. But formal XAI methods also have limitations: they may not scale and hence may be unable to provide explanations for some complex AI models.

Abductive explanations [Shih *et al.*, 2018; Ignatiev *et al.*, 2019b] are the most commonly used formal XAI approach. An abductive explanation is a subset-minimal set of features, which guarantee that if the values of the instance being explained are used for these features, then the decision of the AI system will remain the same. Hence these feature values of the instance are "sufficient" to ensure the decision. For example if a male patient with O+ blood type, height 1.82m and weight 90kg is found to be at risk of a disease by an AI system, an abductive explanation might be that the height and weight are the only required features: thus anyone with height 1.82m and weight 90kg would also be at risk.

While abductive explanations have nice properties and are a valuable tool, they are quite restrictive. The abductive explanation above says nothing about a person with height 1.81m. Recently, *inflated formal explanations* have been proposed [Izza *et al.*, 2024b], which extend abductive explanations to include maximal ranges for features that still guarantee the same decision. An inflated abductive explanation for the disease example may be that any person with height in the range 1.80-2.50m and weight 90-150kg also has the same risk.

When explaining the reason for an AI system decision for a particular instance, there can exist many possible formal abductive explanations; similarly there can exist many possible inflated abductive explanations. Ideally when explaining a decision to a human we would like to give a single explanation. In this work we show how we can compute the *most general abductive explanation*, in some sense the most general explanation of the AI systems behaviour. More precisely, the contributions of this paper are:

1. An implicit hitting set approach to compute most general inflated abductive explanations;
2. Two Boolean Satisfiability (SAT) encodings and an Integer Linear Program (ILP) encoding for generating candidate maximal abductive explanations;
3. Experiments showing that we can in practice create most

general abductive explanations.

2 Preliminaries

2.1 Classification Problems

Let $\mathcal{F}$ be a set of variables called *features*, say $\mathcal{F} = [m]$. Each feature i is equipped with a finite *domain* $\mathbb{D}_i$. In general, when referring to the value of a feature $i \in \mathcal{F}$, we will use a variable x_i, with x_i taking values from $\mathbb{D}_i$. Domains are ordinal that can be integer or real-valued. The *feature space* is defined as $\mathbb{F} = \prod_{i \in \mathcal{F}} \mathbb{D}_i$. The notation $\mathbf{x} = (x_1, \ldots, x_m)$ denotes an arbitrary point in feature space, whilst the notation $\mathbf{v} = (v_1, \ldots, v_m)$ represents a specific point in feature space. A *region* is a set $\mathbb{E} = \mathbb{E}_1 \times \cdots \times \mathbb{E}_m$ consisting of, for each feature i, a non-empty set $\mathbb{E}_i \subseteq \mathbb{D}_i$ of values (it can also be viewed as a function that maps feature i to $\mathbb{E}_i$). A *total classification function* $\kappa : \mathbb{F} \to \mathcal{K}$ where $\mathcal{K} = \{c_1, \ldots, c_K\}$ is a set of *classes* for some $K \geq 2$. For technical reasons, we also require κ not to be a constant function, i.e. there exists at least two points in feature space with differing predictions. An *instance* denotes a pair $(\mathbf{v}, c)$, where $\mathbf{v} \in \mathbb{F}$ and $c \in \mathcal{K}$. We represent a classifier by a tuple $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$. Given the above, an *explanation problem* is a tuple $\mathcal{E} = (\mathcal{M}, (\mathbf{v}, c))$.

2.2 Formal Explainability

Two types of formal explanations have been predominantly studied: abductive [Shih *et al.*, 2018; Ignatiev *et al.*, 2019b] and contrastive [Miller, 2019; Ignatiev *et al.*, 2020]. Abductive explanations (AXps) broadly answer a **Why** question, i.e. *Why the prediction?*, whereas contrastive explanations (CXps) broadly answer a **Why Not** question, i.e. *Why not some other prediction?*. Intuitively, an AXp is a subset-minimal set of feature values $(x_i = v_i)$, at most one for each feature $i \in \mathcal{F}$, that is sufficient to trigger a particular class and satisfy the instance being explained. Similarly, a CXp is a subset-minimal set of features by changing the values of which one can trigger a class different from the target one. More formally, AXps and CXps are defined below.

Given an explanation problem $\mathcal{E} = (\mathcal{M}, (\mathbf{v}, c))$, an *abductive explanation (AXp) of* $\mathcal{E}$ is a subset-minimal set $\mathcal{X} \subseteq \mathcal{F}$ of features which, if assigned the values dictated by the instance $(\mathbf{v}, c)$, are sufficient for the prediction. The latter condition is formally stated, for a set $\mathcal{X}$, as follows:

$$\forall(\mathbf{x} \in \mathbb{F}). \left[\bigwedge\nolimits_{i \in \mathcal{X}} (x_i = v_i)\right] \rightarrow (\kappa(\mathbf{x}) = c) \tag{1}$$

Any subset $\mathcal{X} \subseteq \mathcal{F}$ that satisfies (1), but is not subset-minimal (i.e. there exists $\mathcal{X}' \subset \mathcal{X}$ that satisfies (1)), is referred to as a *weak abductive explanation (Weak AXp)*.

Given an explanation problem, a contrastive explanation (CXp) is a subset-minimal set of features $\mathcal{Y} \subseteq \mathcal{F}$ which, if the features in $\mathcal{F} \setminus \mathcal{Y}$ are assigned the values dictated by the instance $(\mathbf{v}, c)$, then there is an assignment to the features in $\mathcal{Y}$ that changes the prediction. This is stated as follows, for a chosen set $\mathcal{Y} \subseteq \mathcal{F}$:

$$\exists(\mathbf{x} \in \mathbb{F}). \left[\bigwedge\nolimits_{i \in \mathcal{F} \setminus \mathcal{Y}} (x_i = v_i)\right] \wedge (\kappa(\mathbf{x}) \neq c) \tag{2}$$

AXp's and CXp's respect a minimal-hitting set (MHS) duality relationship [Ignatiev *et al.*, 2020]. Concretely, each AXp is an MHS of the CXp's and each CXp is an MHS of the AXp's. MHS duality is a stepping stone for the enumeration of explanations.

Inflated Formal Explanations. An *inflated abductive explanation* (iAXp) of $\mathcal{E} = (\mathcal{M}, (\mathbf{v}, c))$ is a tuple $(\mathcal{X}, \mathbb{E})$, where $\mathcal{X} \subseteq \mathcal{F}$ is an AXp, and $\mathbb{E}$ is a region where (i) $\mathbb{E}_i \subset \mathbb{D}_i$ for each $i \in \mathcal{X}$ and $\mathbb{E}_i = \mathbb{D}_i$ for each $i \in (\mathcal{F} \setminus \mathcal{X})$, (ii) $v_i \in \mathbb{E}_i$ for each $i \in \mathcal{X}$, such that

$$\forall(\mathbf{x} \in \mathbb{F}). \left[\bigwedge\nolimits_{i \in \mathcal{X}} (x_i \in \mathbb{E}_i)\right] \rightarrow (\kappa(\mathbf{x}) = c) \tag{3}$$

and $\mathbb{E}$ is a *maximal set* with properties (i), (ii), and (3), i.e. if $\mathbb{E}'$ is any range that satisfies (i), (ii) and (3), then it is not the case that $\mathbb{E}' \supset \mathbb{E}$.

A *size measure* s is a mononotic function that maps a region $\mathbb{E} \subseteq \mathbb{F}$ to a real number. Consider an iAXp $(\mathcal{X}, \mathbb{E})$ and a size measure s, then $(\mathcal{X}, \mathbb{E})$ is called a *maximum iAXp* if its size score $s(\mathbb{E})$ is a maximum amongst all iAXp's, i.e., if $\mathbb{E}'$ is an iAXp, then $s(\mathbb{E}') \leq s(\mathbb{E})$.

Later, when dealing with tree ensemble models, we will instantiate s to a specific coverage measure on ranges consisting of intervals.

Example 1. *Consider 3 features x_i, $i \in [3]$, representing an individual's blood type, their age, and weight with the domains $\mathbb{D}_1 = \{A, B, AB, O\}$, $\mathbb{D}_2 = [20, 80]$, and $\mathbb{D}_3 = [50, 150]$. Assume the classifier $\kappa(\mathbf{x})$ predicts a high risk of a disease if $x_2 \geq 60$ and $x_3 \geq 80$; otherwise, $\kappa(\mathbf{x})$ predicts the individual to be at low risk of a disease. (Observe that the blood type x_1 is ignored in the decision making process.) Consider an instance $\mathbf{v} = (A, 65, 85)$, which is classified as high risk. The only AXp for this instance is $\mathcal{X} = \{2, 3\}$. Now, consider a simple size measure $s(\mathbb{E}_i) = {|\mathbb{E}_i|}/{|\mathbb{D}_i|}$. While there are multiple ways to inflate $\mathcal{X}$ using this size measure, the* maximum *and, intuitively, the most general iAXp consists of the intervals $\mathbb{E}_2 = [60, 80]$ and $\mathbb{E}_3 = [80, 150]$.*

An *inflated contrastive explanation* (iCXp) is a pair $(\mathcal{Y}, \mathbb{G})$ s.t. $\mathcal{Y}$ is a CXp of $\mathcal{E}$, and $v_i \notin \mathbb{G}_i$ for each feature $i \in \mathcal{Y}$ and $\mathbb{G}_i = \{v_i\}$ for any $i \in (\mathcal{F} \setminus \mathcal{Y})$, such that the following holds:

$$\exists(\mathbf{x} \in \mathbb{F}). \left[\bigwedge\nolimits_{i \in \mathcal{F}} (x_i \in \mathbb{G}_i)\right] \wedge (\kappa(\mathbf{x}) \neq c) \tag{4}$$

Example 2. *Consider again Example 1 and $\mathbf{v} = (A, 65, 85)$ classified as a high risk. If a user is interested in answering* why not *a low risk, a possible CXp to extract is $\mathcal{Y}_1 = \{2\}$. This means that by suitably changing the value of x_2 to some value taken from its entire domain $[20, 80]$, the prediction can be changed to low risk even if features x_1 and x_3 are fixed, resp., to values A and 85. Observe that the freedom of selecting the entire domain $\mathbb{D}_2$ although perfectly valid provides a user with little to no insight on how the prediction can be changed, as it contains values, namely those in $[60, 80]$, that if chosen do not lead to a misclassification. Inflating the CXp $\mathcal{Y}_1$ can be done differently and may result in a valid shrunk interval, say, $\mathbb{G}_2 = [20, 65]$. Intuitively, the most informative iCXp shrinks the domain $\mathbb{D}_2$ into the set $\mathbb{G}_2' = [20, 60)$, while ensuring a misclassification is still achievable. Similarly, another CXp for instance $\mathbf{v}$ is $\mathcal{Y}_2 = \{3\}$, with the most sensible inflation being $\mathbb{G}_3 = [50, 80)$.*

Let us denote the set of all iAXp's for a particular explanation problem $\mathcal{E}$ as $\mathbb{A}(\mathcal{E})$ while the set of all iCXp's will be denoted as $\mathbb{C}(\mathcal{E})$. Similar to the case of traditional abductive and contrastive explanations, minimal hitting set (MHS) duality between iAXp 's and iCXp 's was established in earlier work [Izza *et al.*, 2024b].

Proposition 1. *Given an explanation problem $\mathcal{E}$, each iAXp $(\mathcal{X}, \mathbb{E}) \in \mathbb{A}(\mathcal{E})$ minimally "hits" each iCXp $(\mathcal{Y}, \mathbb{G}) \in \mathbb{C}(\mathcal{E})$ s.t. if feature $i \in \mathcal{F}$ is selected to "hit" iAXp $(\mathcal{X}, \mathbb{E})$ then $\mathbb{G}_i \cap \mathbb{E}_i = \emptyset$, and vice versa.*

Remark 1. *As the previous two examples demonstrate, inflation of an AXp or a CXp can be done in multiple ways resulting in different inflated AXps or CXps, respectively. While many of those inflated explanations are valid, some may have a high degree of redundancy, which is often undesirable in practice. Therefore, the goal of explanation inflation is (i) to expand the intervals included in an iAXp as much as possible and (ii) to shrink the intervals included in an iCXp as much as possible, providing the most general way of (i) ensuring the prediction and (ii) breaking it, respectively. The above proposition establishes a duality between inflated AXps and CXps assuming they are inflated in* the most sensible *way, with no redundancy involved, hence the requirement $\mathbb{G}_i \cap \mathbb{E}_i = \emptyset$.*

Proposition 1 forms the foundation of the algorithm proposed in this work for computing a maximum size inflated abductive explanations inspired by the earlier work in the area of implicit hitting set enumeration [Davies and Bacchus, 2011; Ignatiev *et al.*, 2015; Saikko *et al.*, 2016].

2.3 SAT and MaxSAT

We assume standard definitions for propositional satisfiability (SAT) and maximum satisfiability (MaxSAT) solving [Biere *et al.*, 2021]. The *maximum satisfiability (MaxSAT) problem* is to find a truth assignment that maximizes the number of satisfied propositional formulas in a clausal form (i.e. CNF formula). There are a number of variants of MaxSAT [Biere *et al.*, 2021, Chapters 23 and 24]. We will be mostly interested in *Partial Weighted MaxSAT*, which can be formulated as follows. The input WCNF formula ϕ is of the form $\langle \mathcal{H}, \mathcal{S} \rangle$ where $\mathcal{H}$ is a set of *hard* clauses, which must be satisfied, and $\mathcal{S}$ is a set of *soft* clauses, each with a weight, which represent a preference to satisfy those clauses. Whenever convenient, a soft clause c with weight w will be denoted by (c, w). The *Partial Weighted MaxSAT problem* consists in finding an assignment that satisfies all the hard clauses and maximizes the sum of the weights of the satisfied soft clauses.

2.4 Related Work

Logic-based explanations of ML models have been studied since 2018 [Shih *et al.*, 2018], with recent surveys summarizing the observed progress [Marques-Silva and Ignatiev, 2022; Marques-Silva, 2022; Darwiche, 2023; Marques-Silva, 2024]. However, the study of logic-explanations in AI can at least be traced to the late 1970s and 1980s [Swartout, 1977; Swartout, 1983; Shanahan, 1989].

Inflated explanations are more general explanations than AXps. This has been recognized in the earlier work in model-agnostic explainability (Anchor [Ribeiro *et al.*, 2018]) and in the formal explainability [Izza *et al.*, 2022; Ji and Darwiche, 2023; Yu *et al.*, 2023a; Izza *et al.*, 2023b; Izza *et al.*, 2024b].

Implicit hitting-set algorithms have been used in a wide range of practical domains [Bailey and Stuckey, 2005; Chandrasekaran *et al.*, 2011; Davies and Bacchus, 2011; Previti and Marques-Silva, 2013; Liffiton and Malik, 2013; Ignatiev *et al.*, 2015; Liffiton *et al.*, 2016; Saikko *et al.*, 2016]. In the case of XAI, implicit hitting-set algorithms, mimicking MARCO [Liffiton *et al.*, 2016], have been applied in the enumeration of abductive explanations [Marques-Silva, 2022] but also for deciding feature relevancy [Huang *et al.*, 2023]. Implicit hitting-set algorithms can be viewed as a variant of the general paradigm of counterexample-guided abstraction refinement (CEGAR), which was originally devised in the context of model-checking [Clarke *et al.*, 2003].

Additional recent works on formal explainability for large scale ML models include [Wu *et al.*, 2023; Bassan and Katz, 2023; Izza *et al.*, 2024a; Izza and Marques-Silva, 2024]. Probabilistic explanations are investigated in [Wäldchen *et al.*, 2021; Izza *et al.*, 2023a]. There exist proposals to account for constraints on the inputs [Gorji and Rubin, 2022; Yu *et al.*, 2023c]. Moreover, formal feature attribution explanations are proposed in [Yu *et al.*, 2023b; Biradar *et al.*, 2024; Yu *et al.*, 2024; Létoffé *et al.*, 2025].

3 Tree Ensembles

General TE model. We propose a general model for a tree ensemble (TE) predictor. As shown below, the proposed model serves to represent boosted trees [Friedman, 2001], random forests with majority voting [Breiman, 2001], and random forests with weighted voting (Scikit-learn [Pedregosa and et al., 2011]).

A decision tree T on features $\mathcal{F}$ for classes $\mathcal{K}$ is a binary-branching rooted-tree whose internal nodes are labeled by *split conditions* which are predicates of the form $x_i < d$ for some feature $i \in \mathcal{F}$ and $d \in \mathbb{D}_i$, and leaf nodes labelled by a class $c \in \mathcal{K}$. A path $R_{\mathfrak{n}}$ to a leaf node $\mathfrak{n}$ can be considered as the set of split conditions of nodes visited on the path from root to $\mathfrak{n}$.

A tree ensemble $\mathfrak{T}$ has n decision trees, $T_1, \ldots, T_n$. Each path R_l in each decision tree T_t is assigned a class $cl(l)$ (corresponding to the class of the leaf node it reaches) and a weight w_l. Given a point $\mathbf{v}$ in feature space, and for each decision tree T_t, with $1 \leq t \leq n$, there is a single path $R_{k_{\mathbf{v},t}}$ that is consistent with $\mathbf{v}$ in tree T_t. The index $k_{\mathbf{v},t}$ denotes which path of T_t is consistent with $\mathbf{v}$.

For decision tree T_t, and given $\mathbf{v}$, the picked class is denoted $cl(k_{\mathbf{v},t})$ and the picked weight $w_{k_{\mathbf{v},t}}$. For each class $c \in \mathcal{K}$, the class weight will be computed by,

$$W(c, \mathbf{v}) = \sum\nolimits_{t=1}^{n} \textbf{ if } c = cl(k_{\mathbf{v},t}) \textbf{ then } w_{k_{\mathbf{v},t}} \textbf{ else } 0$$

Finally, the picked class is given by,

$$\kappa(\mathbf{v}) = \mathsf{argmax}_{c \in \mathcal{K}}\{W(c, \mathbf{v})\}$$

Boosted trees. In a boosted tree (BT), and given a point in feature space, class selection works as follows: each tree is pre-assigned a class c (all leaves have the same class c) and returns a weight; the weights of each class are summed up over all trees, and the class with the largest weight is picked.

Random forests with majority voting. In a random forest (RF) with majority voting (RFmv), and given a point in feature space, each decision tree picks a class; the final picked class is the one picked by most trees. Note that to represent an RFmv in our general TE model, we set the weights of the leaf nodes to 1.

Example 3. *Figure 1 shows an example of a RFmv of 3 trees, trained with Scikit-learn on Iris dataset. Consider a sample $\mathbf{v} = (6.0, 3.5, 1.4, 0.2)$ then the counts of votes for classes 1–3 are, resp., $W(c_1, \mathbf{v}) = 2$ (votes of T_2 and T_3), $W(c_2, \mathbf{v}) = 1$ (vote of T_1), and $W(c_3, \mathbf{v}) = 0$. Hence, the predicted class is c_1 ("Setosa") as it has the highest score.*

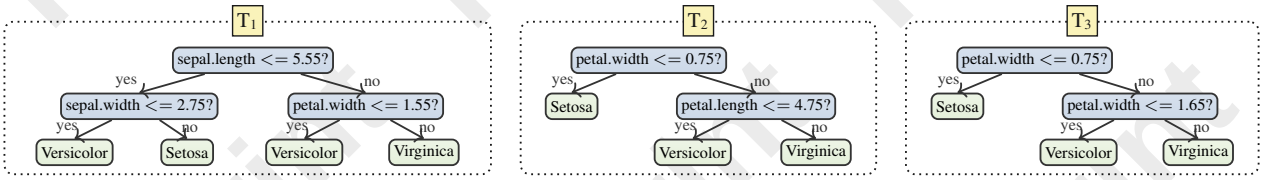

Figure 1: RF with majority votes (RFmv). In the running examples, classes "Setosa", "Versicolor" and "Virginica" are denoted, resp., class c_1, c_2 and c_3, and features *sepal.length*, *sepal.width*, *petal.length* and *petal.width* are ordered, resp., as features 1–4.

Random forests with weighted voting. In an RF with weighted voting (RFwv), and given a point in feature space, each tree picks a weight for a chosen selected class; the final picked class is the class with the largest weight.

Explanation encoding for TEs. Recent advances have applied logic-based approaches to explain tree ensembles (BTs and RFs), where a primary challenge lies in reasoning efficiently over the aggregate output of a large ensemble of decision trees. Hereinafter, we build on the recent propositional encodings of RFs [Izza and Marques-Silva, 2021] and BTs [Ignatiev *et al.*, 2022], which translate the ensemble voting to a SAT formula with cardinality constraints for RFmv or a set of MaxSAT queries expressing pseudo-Boolean constraints for BT.

Roughly, given a tree ensemble $\mathfrak{T}$ defining classifier κ, for each feature $i \in \mathcal{F}$ we can define the set of *split points* $S_i \subset \mathbb{D}_i$ that appear for that feature in all trees, i.e. $S_i = \{d \mid x_i < d$ appears in some tree $T_1, \ldots T_n\}$. Suppose $[s_{i,1}, s_{i,2}, \ldots, s_{i,|S_i|}]$ are the split points in S_i sorted in increasing order. Using this we can define set of disjoint intervals for feature i: $I^i_1 \equiv [\min(\mathbb{D}_i), s_{i,1})$, $I^i_2 \equiv [s_{i,1}, s_{i,2})$, $\ldots$, $I^i_{|S_i|+1} \equiv [s_{i,|S_i|}, \max(\mathbb{D}_i)]$ where $\mathbb{D}_i = I^i_1 \cup I^i_2 \cup \cdots \cup I^i_{|S_i|+1}$. Later, we utilize Boolean variables $[\![x_i < d]\!], d \in S_i$ for each feature $i \in \mathcal{F}$. The key property of a TE is that $\forall \mathbf{x} \in I^1_{e_1} \times I^2_{e_2} \times \cdots \times I^m_{e_m}$ s.t. e_i is defined as the interval index for feature i which includes the value of $\mathbf{v}$, i.e. $v_i \in I^i_{e_i}$, then we have $\kappa(\mathbf{x}) = \kappa(\mathbf{v}) = c$. That is we only need to reason about intervals rather than particular values to explain the behaviour of a tree ensemble. As a result, the AXp (resp. CXp) extraction boils down to finding a subset-minimal subset $\mathcal{X} \in \mathcal{F}$ s.t. (1) holds (resp. (2) holds).

(A unified MaxSAT encoding of TE models — capturing RFwv, RFmv and BT — is outlined in [Izza *et al.*, 2025].)

4 Most General Explanations

Let us assume a tree ensemble $\mathfrak{T}$ and features $i \in \mathcal{F}$ are numerical. Hence, for each feature we obtain a set of intervals $\{I_1, \ldots, I_n\}$ such that $\left(\bigcup_{j=1}^{n} I_j\right) = \mathbb{D}_i$. Assuming the domain $\mathbb{D}_i$ is finite, we can define the *proportion size* of an interval I_j for feature i as $prop_i(I_j) = {}^{|I_j|}/_{|\mathbb{D}_i|}$. For infinite domains $\mathbb{D}_i$ we treat its size as $|\mathbb{D}_i| = \max\{v_i \mid \mathbf{v} \in \mathbb{T}\} - \min\{v_i \mid \mathbf{v} \in \mathbb{T}\}$ where $\mathbb{T}$ is the training set used to train the TE. We can alternatively define the *data proportion size* of an interval I_j for feature i as $data_i(I_j) = {}^{|\{\mathbf{v} \in \mathbb{T} \mid v_i \in I_j\}|}/_{|\mathbb{T}|}$ which give the proportion of the training data that appears in the interval. We can extend both size definitions to work on interval $\mathbb{E}_i$ for feature i in the natural manner: $prop_i(\mathbb{E}_i) = \sum_{I_j \in \mathbb{E}_i} prop_i(I_j)$, and $data_i(\mathbb{E}_i) = \sum_{I_j \in \mathbb{E}_i} data_i(I_j)$.

We can now compute the size of a region $\mathbb{E}$ covered by an inflated explanation $(\mathcal{X}, \mathbb{E})$ as either $prop(\mathbb{E}) = \Pi_{i \in \mathcal{F}} prop_i(\mathbb{E}_i)$ or $data(\mathbb{E}) = \Pi_{i \in \mathcal{F}} data_i(\mathbb{E}_i)$. Either of these can be used to define the size measure $s(\mathbb{E})$.

Note that in the context of explanation inflation, excluding a feature $i \in \mathcal{F}$ from an explanation $(\mathcal{X}, \mathbb{E})$ can be seen as either removing it from $\mathcal{X}$ or, equivalently, as setting $\mathbb{E}_i = \mathbb{D}_i$. For this reason, hereinafter we denote iAXp's as $(\mathcal{X}, \mathbb{E})$ or $\mathbb{E}$ interchangeably. In order to compute maximal size explanations, it will be convenient to compute instead the logarithm of the size of an explanation, thus replacing product by sum. Namely, We define the *feature space coverage* of an explanation $\mathbb{E}$, $FSC_s(\mathbb{E})$ as the log of the size of the explanation using size function s:

$$FSC_s(\mathbb{E}) = \log \prod\nolimits_{i \in \mathcal{F}} s(\mathbb{E}_i) = \sum\nolimits_{i \in \mathcal{F}} (\log s(\mathbb{E}_i)) \quad (5)$$

One key observation is that (5) assumes a uniform distribution over $\mathbb{E}$. However, we emphasize that the (training) data distribution can be seamlessly incorporated into FSC_s by weighting the interval scores $\mathbb{E}_i$ according to their probability distribution.

Maximum inflated explanation (Max-iAXp). Given an explanation problem $(\mathcal{M}, (\mathbf{v}, c))$ and a size metric s, a *maximum inflated abductive explanation* $(\mathcal{X}, \mathbb{E})$ defines the intervals for all the features $i \in \mathcal{X}$, such that the following conditions hold,

$$(\mathcal{X}, \mathbb{E}) \in \mathbb{A}(\mathcal{E}) \quad (6)$$

$$\forall (\mathbb{E}' \subseteq \mathbb{F}).(\mathcal{X}, \mathbb{E}') \notin \mathbb{A}(\mathcal{E}) \lor FSC_s(\mathbb{E}') \leq FCS_s(\mathbb{E}) \quad (7)$$

In other words, $(\mathcal{X}, \mathbb{E})$ is a correct inflated AXp for the explanation problem $\mathcal{E}$, and there is no other correct inflated AXp, which has a region $\mathbb{E}'$ with a larger size under FSC_s.

Implicit hitting dualization algorithm. We devise an algorithm inspired by the *implicit hitting set dualization* paradigm [Bailey and Stuckey, 2005; Chandrasekaran *et al.*, 2011; Davies and Bacchus, 2011; Liffiton and Malik, 2013; Ignatiev *et al.*, 2015; Saikko *et al.*, 2016] for computing maximum inflated AXps. The idea is to use an oracle that searches for explanation candidates $\mathbb{E}$ and a second oracle that checks if $\mathbb{E}$ is an iAXp. The latter solver deals with the encoding of the prediction function of the TE and checks if (3) holds for the candidate $\mathbb{E}$ (in which case it is indeed an iAXp), and reports a counterexample otherwise. The former oracle, i.e., guessing candidate $\mathbb{E}$, encodes a MaxSAT problem where the objective function to maximize is (5) and the hard constraints are representing the intervals $\mathbb{E}_i$. Next, we will describe in detail the encoding that allows computing a minimum hitting set $\mathbb{E}$ of all iCXp's maximizing $FSC_s(\mathbb{E})$.

At each iteration of the algorithm, we obtain a range $\mathbb{E}$ where for each feature i, $\mathbb{E}_i \subseteq \mathbb{D}_i$, $\mathbb{E}_i = \bigcup_{l \leq j \leq u} I^i_j$ (s.t.

Algorithm 1 Computing maximum inflated AXp for TE

Input: Expl. prob. $\mathcal{E} = (\mathcal{M}, (\mathbf{v}, c))$; WCNF $\langle \mathcal{H}, \mathcal{S} \rangle$
Output: One Max-iAXp $(\mathcal{X}, \mathbb{E})$

repeat
  $(\mu, \mathbf{s}) \leftarrow \mathsf{MaxSAT}(\mathcal{H}, \mathcal{S})$
  $\mathbb{E} \leftarrow \{\bigcup_{l \leq j \leq u} I^i_j \,|\, \forall (y^i_{l,u} \in \mathcal{S}).\mu(y^i_{l,u}) = 1\}$
  $\mathcal{X} \leftarrow \{i \in \mathcal{F} \,|\, |\mathbb{E}_i| < |\mathbb{D}_i|\}$
  $\mathsf{hasCEx} \leftarrow \neg\mathsf{WiAXp}(\mathcal{E}; \mathcal{X}, \mathbb{E})$
  if $\mathsf{hasCEx} =$ **true then**
    $(\mathcal{Y}, \mathbb{G}) \leftarrow \mathsf{FindiCXp}(\mathcal{E}; \mathcal{F}, \mathbb{E})$
    $\mathcal{H} \leftarrow \mathcal{H} \cup \mathsf{newBlockCl}(\mathcal{Y}, \mathbb{G})$
until $\mathsf{hasCEx} =$ **false**
return $(\mathcal{X}, \mathbb{E})$

$I^i_j \equiv [s_{i,l}, s_{i,u})$) and $v_i \in \mathbb{E}_i$. We construct $\mathbb{E}$ and check if it is an iAXp, i.e. there is no adversarial example.

Observe that it is sufficient to show $\mathbb{E}$ is not a Weak iAXp to prove the existence of a CXp in $\mathbb{E}$ — (4) holds — by using any encoding of the TE to find a solution in $\mathbf{x} \in \mathbb{E}$ where $\kappa(\mathbf{x}) \neq c$. This is achieved by fixing the Booleans of the TE encoding implied by the candidate explanation $\mathbb{E}$, i.e. $[\![x_i < d]\!] = \top$ when $d \leq \mathsf{sup}(\mathbb{E}_i)$ and $[\![x_i < d]\!] = \bot$ when $d \geq \mathsf{inf}(\mathbb{E}_i)$, and maximizing $\mathcal{S}_{c,c'}$ for all $c' \neq c$. If we find a solution with $\mathcal{S}_{c,c'} > 0$ this is an adversarial example. If the oracle encoding the TE decides (3) to be false, that is it finds an adversarial example, then we reduce the adversarial example into a CXp $\mathcal{Y} \subseteq \mathcal{F}$ (or inflated CXp $(\mathcal{Y}, \mathbb{G})$ and block it in the hitting set oracle such that the next candidate hits this CXp. Otherwise, the oracle reports (3) to be true, subsequently we conclude that $\mathbb{E}$ meets the conditions (6)–(7). Pseudo-code for the algorithm is shown in Algorithm 1.

A consequence of the MHS duality between iAXp's and iCXp's [Izza *et al.*, 2024b] is that the checker oracle reports a new candidate to be an iAXp and the algorithm terminates before the hitting set oracle exhausts hitting sets to enumerate. Moreover, it is clear that the hitting set oracle finds the largest unblocked interval for each iteration that maximizes the score of the objective function $FSC_s(\mathbb{E})$. Also, note that at the first iteration, as the oracle does not know anything about the model (no blocked subdomains yet), it will cover all the feature space (all features are free). Finally, as a simple improvement of the approach, one can compute initially iCXp's of size 1 and block those adversarial regions at the beginning, before entering the hitting set enumeration loop.

Generally, as Algorithm 1 exploits the ideas of implicit hitting set dualization, it shares the worst-case exponential number of iterations with the rest of this paradigm's instantiations. Note that the number of iterations here grows exponentially not only with respect to the number of features as in the standard smallest-size AXp extraction [Ignatiev *et al.*, 2019b; Ignatiev *et al.*, 2020] but also on the feature domain size.

Naive MaxSAT encoding. What remains to explain is how we encode the candidate enumeration process as a MaxSAT problem $\langle \mathcal{H}, \mathcal{S} \rangle$, and how we create the blocking clauses.

We create Boolean variables $y^i_{l,u}$ to represent the interval $\mathbb{E}_i = \cup_{l \leq j \leq u} I^i_j$. Note that we can restrict to cases where $l \leq e_i \leq u$ since any iAXp must include the interval of the example being explained e_i for each feature i. Importantly, we can define the log of the weight of each interval $w^i_{l,u} = \log(\sum_{l \leq j \leq u} s(I^i_k))$. In order to use MaxSAT we need to fix the entire interval for feature i so that the resulting objective is a sum. Hence we generate $O(|S_i|^2)$ Booleans for each feature i.

Notation	Role
$y^i_{l,u}$	Boolean variable encoding an interval $\mathbb{E}_i = \cup_{l \leq j \leq u} I^i_j$ of feature i
$[\![\mathbb{E}_i \not\supseteq I^i_{a_i,b_i}]\!]$	Boolean variable to test if a sub-interval $I^i_{a_i,b_i}$ is (not) in $\mathbb{E}_i$
$[\![l_i \geq s_{i,j}]\!]$	Boolean variable encoding the lower bound of an interval $\mathbb{E}_i$
$[\![u_i \leq s_{i,j}]\!]$	Boolean variable encoding the upper bound of an interval $\mathbb{E}_i$
$w^i_{l,u}$	Denotes the weight of an interval $\mathbb{E}_i = \cup_{l \leq j \leq u} I^i_j$ of feature i
$FSC_s(\mathbb{E})$	Explanation coverage function $FSC_s(\mathbb{E}) = \sum_{i \in \mathcal{F}} w^i_{l,u} y^i_{l,u}$ to maximize

Table 1: Outline of variables/notations declared in the encodings.

The only hard clauses $\mathcal{H}$ of the MaxSAT model encode the cardinality constraints

$$\forall_{i \in \mathcal{F}} \sum\nolimits_{1 \leq l \leq e_i \leq u \leq |S_i|+1} y^i_{l,u} \leq 1 \tag{8}$$

enforcing that for each feature at most one interval covering the instance to be explained is selected. The soft clauses $\mathcal{S}$ of the MaxSat model are $(y^i_{l,u}, w^i_{l,u})$, indicating our objective is to maximise $\sum_{i \in \mathcal{F}} w^i_{l,u} y^i_{l,u} = FSC_s(\mathbb{E})$.

Example 4. *Consider the feature $i = 4$, petal.width of the RF of Figure 1, then $S_4 = \{0.75, 1.55, 1.65\}$ and we assume $\mathbb{D}_4 = [0, 3]$. This defines 4 intervals $I^4_1 = [0, 0.75)$, $I^4_2 = [0.75, 1.55)$, $I^4_3 = [1.55, 1.65)$, $I^4_4 = [1.65, 3]$. If we are explaining the instance from Example 3 then the interval where the instance sits is $e_4 = 1$ since $petal.width = 0.2$. We construct Booleans $y^4_{1,1}$, $y^4_{1,2}$, $y^4_{1,3}$, $y^4_{1,4}$ representing $\mathbb{E}_4$ as one of $[0, 0.75)$, $[0, 1.55)$, $[0, 1.65)$, $[0, 3]$. We require at most one $y^4_{l,u}$ to be true, and give them weights defined by the interval weight $FSC_s(I^4_{l,u})$.*

A solution to the MaxSAT model assigns exactly one interval to each feature, thus defining a candidate explanation $\mathbb{E}$. Optimality of the MaxSAT solution means it will be the candidate explanation of largest total size.

We use the TE MaxSAT model to determine if $\mathbb{E}$ include a counterexample. If this fails then $\mathbb{E}$ is an iAXp, and by the maximality of size it is the maximum inflated AXp. Otherwise, we have a counterexample which we need to exclude from future intervals.

Assume the TE model, given a candidate explanation $\mathbb{E}$ returns a counterexample, then we have a solution on the $[\![x_i < d]\!]$ variables, falling within $\mathbb{E}$ where $\kappa(\mathbf{w}) \neq c$ for all $\mathbf{w}$ which satisfy these bounds. We can extract a counterexample range $\mathbb{G}$ from the solution, where $\mathbb{G}_i = I^i_{a_i,b_i}$ given by $a_i = \max(\{j + 1 \mid j \in 1..|S_i|, [\![x_i < s_{i,j}]\!] = \bot\} \cup \{1\})$ and $b_i = \min(\{j \mid j \in 1..|S_i| \mid [\![x_i < s_{i,j}]\!] = \top\} \cup \{|S_i| + 1\})$. We now add clauses to the MaxSAT model to prevent selecting candidate explanations that would cover this adversarial example. Note that if $a_i \leq e_i \leq b_i$ then we cannot differentiate the counterexample from the instance being explained using feature i. Hence we want to express the constraint that $\mathbb{E}_i \not\supseteq I^i_{a_i,b_i}$ for at least one $i \in \mathcal{F}$ where it is not the case that $a_i \leq e_i \leq b_i$.

We can express the condition $\mathbb{E}_i \not\supseteq I^i_{a_i,b_i}$ as the conjunction $\bigwedge_{l \leq a_i \wedge b_i \leq u} \neg y^i_{l,u}$, since we can no longer choose an interval, which covers the interval of the counterexample. The

"blocking clause" we add to the MaxSAT model is then the disjunction $\bigvee_{i\in\mathcal{Y},\neg(a_i\leq e_i\leq b_i)}\bigwedge_{l\leq a_i\wedge b_i\leq u}\neg y^i_{l,u}$. To translate this into clausal form, we add auxiliary Boolean variables $[\![\mathbb{E}_i \not\supseteq I^i_{a_i,b_i}]\!]$ and the clausal encoding of: $[\![\mathbb{E}_i \not\supseteq I^i_{a_i,b_i}]\!] \leftrightarrow \bigwedge_{l\leq a_i\wedge b_i\leq u}\neg y^i_{l,u}$ and $\bigvee_{i\in\mathcal{Y},\neg(a_i\leq e_i\leq b_i)}[\![\mathbb{E}_i \not\supseteq I^i_{a_i,b_i}]\!]$ to the MaxSAT model.

Example 5. *Suppose for explaining the instance of Example 3 we find counterexample* $\mathbb{G}$ *corresponding to sample* $\mathbf{w} = (6.0, 3.5, 1.4, 0.8)$ *which is predicted as Versicolor and only differs in the* $petal.width$ *value. For this example* $[\![x_4 < 0.75]\!] = \bot$ *and* $[\![x_4 < 1.55]\!] = \top$*, so* $\mathbb{G}_4 = I^4_{2,2}$*. We enforce new variable* $[\![\mathbb{E}_4 \not\supseteq I^4_{2,2}]\!] \leftrightarrow (\neg y^4_{1,2}\wedge\neg y^4_{1,3}\wedge\neg y^4_{1,4})$*, and add the blocking clause* $[\![\mathbb{E}_4 \not\supseteq I^4_{2,2}]\!]$ *forcing it to hold. We can no longer choose these intervals when searching for iAXp candidate* $\mathbb{E}$*.*

Note that we only add definitions for Booleans $[\![\mathbb{E}_i \not\supseteq I^i_{a_i,b_i}]\!]$ *on demand*, that is when they occur in a counterexample. This prevents us creating a large initial model, where many of these auxiliary Booleans may never be required.

Bounds-based MaxSAT encoding. The above MaxSAT encoding creates large cardinality encodings (8), and may generate many Booleans of the form $[\![\mathbb{E}_i \not\supseteq I^i_{a_i,b_i}]\!]$ in order to block counterexamples. We can do this much more succinctly using a bounds representation.

Here we introduce Boolean variables to represent the lower and upper bounds of the interval $\mathbb{E}_i$: $\{[\![l_i \geq d]\!] \mid d \in S_i, d \leq \min(I^i_{e_i})\}$ and $\{[\![u_i < d]\!] \mid d \in S_i, d \geq \max(I^i_{e_i})\}$. Note that we do not need to represent bounds which exclude the explained instance value for this feature i, v_i. We add to $\mathcal{H}$ the implications $[\![l_i \geq s_{i,j+1}]\!] \rightarrow [\![l_i \geq s_{i,j}]\!], 1 \leq j < e_i$ and $[\![u_i < s_{i,j}]\!] \rightarrow [\![u_i < s_{i,j+1}]\!], e_i \leq j \leq |S_i|$.

We define the interval variables $y^i_{l,u}$ using clauses in $\mathcal{H}$ encoding

$$y^i_{l,u} \quad \leftrightarrow \quad [\![l_i \geq s_{i,l-1}]\!] \wedge \neg[\![l_i \geq s_{i,l}]\!] \wedge \neg[\![u_i < s_{i,u}]\!] \wedge [\![u_i < s_{i,u+1}]\!]$$

where $[\![l_i \geq s_{i,0}]\!]$ and $[\![u_i < s_{i,|S_i|+1}]\!]$ are both treated as $\top$. We use the same soft clauses to define the objective as in the Naive encoding.

Example 6. *Encoding feature* $petal.width$ *we generate only the bounds* $[\![u_4 < 0.75]\!]$*,* $[\![u_4 < 1.55]\!]$*,* $[\![u_4 < 1.65]\!]$ *and implications* $[\![u_4 < 0.75]\!] \rightarrow [\![u_4 < 1.55]\!]$ *and* $[\![u_4 < 1.55]\!] \rightarrow [\![u_4 < 1.65]\!]$*, and* $y^4_{1,1} \leftrightarrow [\![u_4 < 0.75]\!]$*,* $y^4_{1,2} \leftrightarrow \neg[\![u_4 < 0.75]\!] \wedge [\![u_4 < 1.55]\!]$*,* $y^4_{1,3} \leftrightarrow \neg[\![u_4 < 1.55]\!] \wedge [\![u_4 < 1.65]\!]$*, and* $y^4_{1,4} \leftrightarrow \neg[\![u_4 < 1.65]\!]$*.*

If the test whether $\mathbb{E}$ is an iAXp fails, we are given an (inflated) CXp $(\mathcal{Y}, \mathbb{G})$. Adding a blocking clause to the bounds-based MaxSAT encoding is much simpler. Suppose $\mathbb{G}_i = I^i_{a_i,b_i}$ where $e_i \notin [a_i, b_i]$ then if $e_i < a_i$ we can hit the iCXp for feature i by enforcing $[\![u_i \leq s_{i,a_i-1}]\!]$, similarly if $e_i > b_i$ we can hit the iCXp for feature i by enforcing $[\![l_i \geq s_{i,b_i}]\!]$. The blocking clause is the disjunction of these literals for all $i \in \mathcal{F}$ where $e_i \notin [a_i, b_i]$.

Example 7. *Given the same counter example as in Example 5 we have* $a_4 = b_4 = 2$ *and* $e_4 = 1$*, so the blocking clause we add for the bounds model is simply* $[\![u_4 < 0.75]\!]$*.*

MIP encoding. We can easily rewrite the bounds based MaxSAT encoding to a mixed integer linear program (MIP model). All the Boolean variables become 0-1-integer variables. The implications become simple inequalities, e.g. $[\![l_i \geq s_{i,j+1}]\!] \rightarrow [\![l_i \geq s_{i,j}]\!]$ becomes $[\![l_i \geq s_{i,j+1}]\!] \leq [\![l_i \geq s_{i,j}]\!]$. The definitions of the interval variables $y^i_{l,u}$ become a series of inequalities where negation $\neg l$ is modelled by $1 - l$. For example the definition of $y^i_{l,u}$ becomes

$$\begin{aligned} y^i_{l,u} &\leq [\![l_i \geq s_{i,l-1}]\!] \\ y^i_{l,u} &\leq 1 - [\![l_i \geq s_{i,l}]\!] \\ y^i_{l,u} &\leq 1 - [\![u_i < s_{i,u}]\!] \\ y^i_{l,u} &\leq [\![l_i \geq s_{i,u+1}]\!] \\ y^i_{l,u} + 3 &\geq [\![l_i \geq s_{i,l-1}]\!] + 1 - [\![l_i \geq s_{i,l}]\!] + \\ &\quad\; 1 - [\![u_i < s_{i,u}]\!] + [\![l_i \geq s_{i,u+1}]\!] \end{aligned}$$

The objective in the MIP model is to maximize $\sum_{i\in\mathcal{F},1\leq l\leq e_i\leq u\leq|S_i|+1} w^i_{l,u} \cdot y^i_{l,u}$, which is completely analogous to the MaxSAT encoding. The blocking clauses are also representable as a simple linear inequality.

5 Experiments

This section presents a summary of empirical assessment of computing maximum inflated abductive explanations for tree ensembles — the case study of RFmv and BT — trained on some of the widely studied datasets. Our evaluation aims to investigate the following research questions:

- **RQ1**: Are *hypercubes* representing Max-iAXp explanations much larger than iAXp's on real world benchmarks?
- **RQ2**: Are the proposed logical encodings scale for practical RFs and BTs?
- **RQ3**: Does our algorithm converge quickly to deliver the optimal explanation?

Experimental Setting. The experiments are conducted on Intel Core i5-10500 3.1GHz CPU with 16GByte RAM running Ubuntu 22.04 LTS. A time limit for each tested instance is fixed to 900 seconds (i.e. 15 minutes); whilst the memory limit is set to 4 GByte.

The assessment of TEs (RFs and BTs) is performed on a selection of publicly available datasets, which originate from UCI ML Repository [UCI, 2020] and Penn ML Benchmarks [Olson *et al.*, 2017] — in total 12 datasets. Note that datasets are fully numerical as we are interested in assessing the proposed FSC coverage metric, which discriminate the intervals given their size. For categorical data, the FSC scores are the same for all domain values, then it is pointless in our empirical evaluation to include them. When training RFs, we used Scikit-learn [Pedregosa and et al., 2011] and for BTs we applied XGBoost [Chen and Guestrin, 2016].

The proposed approach is implemented in `RFxpl`[1] and `XReason`[2] Python packages. The `PySAT` toolkit [Ignatiev *et al.*, 2018; Ignatiev *et al.*, 2024] is used to instrument SAT or/and MaxSAT oracle calls. RC2 [Ignatiev *et al.*, 2019a] MaxSAT solver, which is implemented in PySAT, is applied to all MaxSAT encodings (for the TE operation and hitting set dualization). Moreover, Gurobi [Gurobi Optimization, LLC, 2023] is applied to instrument MIP oracle calls using its Python interface.

[1]https://github.com/izzayacine/RFxpl
[2]https://github.com/alexeyignatiev/xreason

Dataset	(m, K)	iAXp			Max-iAXp			Naive MxS		Bnd MxS		Bnd MIP		Cov. Ratio	
		Len	Cov%	Time	Len	Cov%	calls	Time_{rc2}	TO_{rc2}	Time_{rc2}	TO_{rc2}	Time_{grb}	TO_{grb}	avg	max
ann-thyroid	(21 3)	1.9	2.59	0.41	1.9	8.94	7.3	23.21	0	10.28	0	2.19	0	**3.446**	524.711
appendicitis	(7 2)	5.2	5.30	0.16	4.2	18.44	62.7	282.10	19	172.11	11	97.23	0	**3.478**	4676.519
banknote	(4 2)	2.3	11.84	0.12	2.2	24.11	8.3	428.67	13	266.35	1	20.98	0	**2.037**	786.040
breast-cancer	(9 2)	3.6	15.39	0.12	3.7	23.27	12.0	0.22	0	0.19	0	0.32	0	1.512	5.500
bupa	(6 2)	4.7	0.39	0.15	4.6	4.20	18.6	58.62	2	38.54	0	13.02	0	**10.662**	3128.349
car	(6 4)	1.6	52.42	0.21	1.6	53.61	1.9	0.14	0	0.21	0	0.14	0	1.023	4.167
ecoli	(7 5)	3.9	3.19	0.87	3.8	7.15	19.7	229.92	10	170.17	2	54.23	0	**2.238**	6350.058
haberman	(3 2)	1.6	6.54	0.08	1.7	8.70	5.0	0.46	0	0.41	0	0.89	0	1.330	11.011
iris	(4 3)	2.1	9.65	0.20	2.1	9.85	8.1	0.45	0	0.65	0	0.63	0	1.021	2.302
lupus	(3 2)	1.4	19.93	0.08	1.8	26.58	6.4	5.99	0	0.69	0	0.94	0	1.334	114.821
new-thyroid	(3 2)	3.1	9.98	0.28	3.2	12.62	16.6	199.28	2	26.35	0	8.56	0	1.263	6048.512
wine-recog	(13 3)	11.2	1.51	0.27	4.3	3.49	140.7	88.03	22	366.51	20	196.06	0	**2.318**	364.379

Table 2: Performance evaluation of computing maximum inflated AXp's for RFmv. For each dataset, we randomly pick 25 instances to test. **Max-iAXp** reports the average explanation length, coverage (in %) and MaxSAT oracle calls in Algorithm 1. **Naive MxS**, **Bnd MxS** and **Bnd MIP** report the average runtime for computing Max-iAXp and total timeout tests, resp., for naive MaxSAT, Bounds-based MaxSAT and MIP encodings. Column **Cov. Ratio** reports, resp., the average and maximum ratio between domain coverage of **Max-iAXp** and **iAXp**. Coverage ratios higher than a factor of 2 are highlighted in bold text and Max-iAXp coverages greater than 10% are highlighted in grey.

RQ1. Table 2 shows the coverage size of Max-iAXp and baseline iAXp explanations, for the case study of RFmv tree ensembles. For a more fine-grained evaluation, we compare the scalability of the proposed logical encodings, i.e. naive and bound-based MaxSAT encodings as well as MIP encoding in Table 2. As can be observed from Table 2, the average domain coverage of Max-iAXp's is at least twice and up to 10 times, larger than the average coverage of iAXp's for a half of datasets while it is fairly superior for the remaining datasets.

Interestingly, we observe that the maximum ratio between feature space coverage offered by Max-iAXp and iAXp can be extremely high for the majority of benchmarks (e.g. 3128 to 6048 times wider in *appendicitis*, *ecoli*, *bupa* and *new-thyroid*) with a few exceptions on smaller datasets (e.g. *breast-cancer*, *car* and *iris*, resp. showing a ratio of 5.5, 4.1 and 2.3). Furthermore, the average lengths of iAXp and Max-iAXp remain consistently close, even when there is a large gap in coverage scores, and very succinct w.r.t. input data (e.g. $\sim 6.6\%$ in *ann-thyroid*).

RQ2 & RQ3. Performance-wise, we observe a substantial advantage of MIP oracle (Gurobi) over MaxSAT (RC2) despite the fact they use essentially the same formulation of the problem. Although there may be various reasons for this phenomenon, the hitting set problem seems to be more amenable to MIP solvers as they can take advantage of multiple efficient optimization procedures, including linear relaxation, branch-and-bound methods augmented with cutting planes resolution as well as both primal and dual reasoning. On the contrary, RC2 is a core-guided MaxSAT solver designed to handle optimization problems that are inherently Boolean, which is the case for the TE encoding we used in our work where RC2 shines. As the results demonstrate, our approach empowered with Gurobi is able to deliver explanations within $\sim$33 sec for the average runtime for all the considered datasets (and a min. (resp. max.) of 0.14 sec ($\sim$3 min)), and able to terminate on all tests, whilst RC2 gets timed out on 4 datasets (particularly for *wine-recog* in 20 out of 25 instances).

Although computing iAXp's in general tends to be faster than Max-iAXp (which is not surprising), applying our approach to computing Max-iAXp is worthwhile when most general explanations are of concern. This is underscored by the fact that the approach is shown to scale. Finally, the improved bounds-based variant of the approach (both with MaxSAT and MIP) is clearly superior to the naive propositional encoding in terms of the overall performance.

Dataset	iAXp		Max-iAXp			Cov. Ratio	
	Cov%	Time	Cov%	calls	Time_{grb}	avg	max
ann-thyroid	1.34	0.27	17.77	2.0	1.56	**13.27**	59.4
appendicitis	6.59	0.09	24.26	10.4	0.90	**3.68**	587.7
banknote	18.21	1.13	21.33	5.5	15.86	1.17	22.3
breast-cancer	0.38	1.14	2.21	17.9	10.61	**5.81**	131.6
bupa	0.49	1.26	2.78	20.6	18.92	**5.63**	98379.4
car	26.54	0.14	26.91	2.2	0.18	1.01	22.5
ecoli	2.92	6.88	5.29	17.8	83.84	1.81	743.8
haberman	1.40	0.56	2.71	6.0	3.30	1.93	36.2
iris	20.63	0.13	22.29	3.2	0.38	1.08	22.5
lupus	16.56	0.07	19.47	3.6	0.28	1.18	7.7
new-thyroid	4.63	0.41	7.39	9.7	2.73	1.60	246.7
wine-recog	6.18	0.26	14.02	16.1	3.80	**2.27**	212.8

Table 3: Performance evaluation of computing Max-iAXp's of BTs.

Table 3 reports the results of Max-iAXp for BTs focusing solely on MIP encoding. Similarly to RFs, we observe large Max-iAXp explanation coverage for most of the benchmarks, where 8/12 datasets show 12% to 53% coverage of the total input space. Additionally, we observe a net superiority of average coverage of Max-iAXp over iAXp with a factor of 2 up to 13 for 5/12 datasets. Overall, these empirical results validate the effectiveness and explanatory wide-ranging of our framework in explaining TEs.

6 Conclusions

This paper takes a step further in the quest of finding most succinct and general explanations for (complex) ML models. This is achieved by formalizing the concept of *maximum inflated abductive explanation* (Max-iAXp), which can be considered as the most general abductive explanations. Furthermore, the paper proposes an elegant approach for the rigorous computation of maximum inflated explanations, and demonstrates the broader coverage of such explanations in comparison with maximal (subset) inflated abductive explanations. The experimental results validate the practical interest of computing *maximum* iAXp explanations.

Acknowledgments

This work was supported in part by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme, by the Spanish Government under grant PID2023-152814OB-I00 and ICREA starting funds, and by the Australian Research Council through the OPTIMA ITTC IC200100009.

Finally, we thank Rayman Tang for the discussions at an early stage of this work and the reviewers for their constructive feedbacks.

References

[Bailey and Stuckey, 2005] James Bailey and Peter J. Stuckey. Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In *PADL*, pages 174–186, 2005.

[Bassan and Katz, 2023] Shahaf Bassan and Guy Katz. Towards formal XAI: formally approximate minimal explanations of neural networks. In *TACAS*, pages 187–207, 2023.

[Biere *et al.*, 2021] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability - Second Edition*. 2021.

[Biradar *et al.*, 2024] Gagan Biradar, Yacine Izza, Elita A. Lobo, Vignesh Viswanathan, and Yair Zick. Axiomatic aggregations of abductive explanations. In *AAAI*, pages 11096–11104, 2024.

[Breiman, 2001] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

[Chandrasekaran *et al.*, 2011] Karthekeyan Chandrasekaran, Richard M. Karp, Erick Moreno-Centeno, and Santosh S. Vempala. Algorithms for implicit hitting set problems. In *SODA*, pages 614–629, 2011.

[Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *KDD*, pages 785–794, 2016.

[Clarke *et al.*, 2003] Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement for symbolic model checking. *J. ACM*, 50(5):752–794, 2003.

[Darwiche, 2023] Adnan Darwiche. Logic for explainable AI. In *LICS*, pages 1–11, 2023.

[Davies and Bacchus, 2011] Jessica Davies and Fahiem Bacchus. Solving MAXSAT by solving a sequence of simpler SAT instances. In *CP*, pages 225–239, 2011.

[Friedman, 2001] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[Gorji and Rubin, 2022] Niku Gorji and Sasha Rubin. Sufficient reasons for classifier decisions in the presence of domain constraints. In *AAAI*, pages 5660–5667, 2022.

[Gurobi Optimization, LLC, 2023] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[Huang *et al.*, 2023] Xuanxiang Huang, Yacine Izza, and João Marques-Silva. Solving explainability queries with quantification: The case of feature relevancy. In *AAAI*, pages 3996–4006, 2023.

[Ignatiev *et al.*, 2015] Alexey Ignatiev, Alessandro Previti, Mark H. Liffiton, and Joao Marques-Silva. Smallest MUS extraction with minimal hitting set dualization. In *CP*, pages 173–182, 2015.

[Ignatiev *et al.*, 2018] Alexey Ignatiev, António Morgado, and Joao Marques-Silva. PySAT: A python toolkit for prototyping with SAT oracles. In *SAT*, pages 428–437, 2018.

[Ignatiev *et al.*, 2019a] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. RC2: an efficient MaxSAT solver. *J. Satisf. Boolean Model. Comput.*, 11(1):53–64, 2019.

[Ignatiev *et al.*, 2019b] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. Abduction-based explanations for machine learning models. In *AAAI*, pages 1511–1519, 2019.

[Ignatiev *et al.*, 2020] Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and Joao Marques-Silva. From contrastive to abductive explanations and back again. In *AIxIA*, pages 335–355, 2020.

[Ignatiev *et al.*, 2022] Alexey Ignatiev, Yacine Izza, Peter J. Stuckey, and João Marques-Silva. Using MaxSAT for efficient explanations of tree ensembles. In *AAAI*, pages 3776–3785, 2022.

[Ignatiev *et al.*, 2024] Alexey Ignatiev, Zi Li Tan, and Christos Karamanos. Towards universally accessible SAT technology. In *SAT*, pages 4:1–4:11, 2024.

[Ignatiev, 2020] Alexey Ignatiev. Towards trustable explainable AI. In *IJCAI*, pages 5154–5158, 2020.

[Izza and Marques-Silva, 2021] Yacine Izza and Joao Marques-Silva. On explaining random forests with SAT. In *IJCAI*, pages 2584–2591, 2021.

[Izza and Marques-Silva, 2024] Yacine Izza and Joao Marques-Silva. Efficient contrastive explanations on demand. *CoRR*, abs/2412.18262, 2024.

[Izza *et al.*, 2022] Yacine Izza, Alexey Ignatiev, and João Marques-Silva. On tackling explanation redundancy in decision trees. *J. Artif. Intell. Res.*, 75:261–321, 2022.

[Izza *et al.*, 2023a] Yacine Izza, Xuanxiang Huang, Alexey Ignatiev, Nina Narodytska, Martin C. Cooper, and João Marques-Silva. On computing probabilistic abductive explanations. *Int. J. Approx. Reason.*, 159:108939, 2023.

[Izza *et al.*, 2023b] Yacine Izza, Alexey Ignatiev, Peter J. Stuckey, and João Marques-Silva. Delivering inflated explanations. *CoRR*, 2023.

[Izza *et al.*, 2024a] Yacine Izza, Xuanxiang Huang, Antonio Morgado, Jordi Planes, Alexey Ignatiev, and Joao Marques-Silva. Distance-Restricted Explanations: Theoretical Underpinnings & Efficient Implementation. In *KR*, pages 475–486, 2024.

[Izza *et al.*, 2024b] Yacine Izza, Alexey Ignatiev, Peter J. Stuckey, and João Marques-Silva. Delivering inflated explanations. In *AAAI*, pages 12744–12753, 2024.

[Izza *et al.*, 2025] Yacine Izza, Alexey Ignatiev, Sasha Rubin, Joao Marques-Silva, and Peter J. Stuckey. Most general explanations of tree ensembles (extended version). *CoRR*, abs/2505.10991, 2025.

[Ji and Darwiche, 2023] Chunxi Ji and Adnan Darwiche. A new class of explanations for classifiers with non-binary features. In *JELIA*, pages 106–122, 2023.

[Létoffé *et al.*, 2025] Olivier Létoffé, Xuanxiang Huang, and João Marques-Silva. Towards trustable SHAP scores. In *AAAI*, pages 18198–18208, 2025.

[Liffiton and Malik, 2013] Mark H. Liffiton and Ammar Malik. Enumerating infeasibility: Finding multiple muses quickly. In *CPAIOR*, pages 160–175, 2013.

[Liffiton *et al.*, 2016] Mark H. Liffiton, Alessandro Previti, Ammar Malik, and João Marques-Silva. Fast, flexible MUS enumeration. *Constraints An Int. J.*, 21(2):223–250, 2016.

[Lundberg and Lee, 2017] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, pages 4765–4774, 2017.

[Marques-Silva and Huang, 2024] João Marques-Silva and Xuanxiang Huang. Explainability is not a game. *Commun. ACM*, 67(7):66–75, 2024.

[Marques-Silva and Ignatiev, 2022] João Marques-Silva and Alexey Ignatiev. Delivering trustworthy AI through formal XAI. In *AAAI*, pages 12342–12350, 2022.

[Marques-Silva, 2022] João Marques-Silva. Logic-based explainability in machine learning. In *Reasoning Web*, pages 24–104, 2022.

[Marques-Silva, 2024] João Marques-Silva. Logic-based explainability: Past, present and future. In *ISoLA*, volume 15222, pages 181–204, 2024.

[Miller, 2019] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019.

[Molnar, 2020] Christoph Molnar. *Interpretable machine learning*. 2020.

[Olson *et al.*, 2017] Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. PMLB: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10(1):36, 2017.

[Pedregosa and et al., 2011] Fabian Pedregosa and et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Previti and Marques-Silva, 2013] Alessandro Previti and Joao Marques-Silva. Partial MUS enumeration. In *AAAI*, pages 818–825, 2013.

[Ribeiro *et al.*, 2016] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *KDD*, pages 1135–1144, 2016.

[Ribeiro *et al.*, 2018] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, pages 1527–1535, 2018.

[Rudin *et al.*, 2022] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022.

[Rudin, 2019] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[Saikko *et al.*, 2016] Paul Saikko, Johannes Peter Wallner, and Matti Jarvisalo. Implicit hitting set algorithms for reasoning beyond NP. In *KR*, pages 104–113, 2016.

[Samek *et al.*, 2019] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller, editors. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019.

[Samek *et al.*, 2021] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proc. IEEE*, 109(3):247–278, 2021.

[Shanahan, 1989] Murray Shanahan. Prediction is deduction but explanation is abduction. In *IJCAI*, pages 1055–1060, 1989.

[Shih *et al.*, 2018] Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining bayesian network classifiers. In *IJCAI*, pages 5103–5111, 2018.

[Slack *et al.*, 2020] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling LIME and SHAP: adversarial attacks on post hoc explanation methods. In *AIES*, pages 180–186, 2020.

[Swartout, 1977] William R. Swartout. A digitalis therapy advisor with explanations. In *IJCAI*, pages 819–825, 1977.

[Swartout, 1983] William R. Swartout. XPLAIN: A system for creating and explaining expert consulting programs. *Artif. Intell.*, 21(3):285–325, 1983.

[UCI, 2020] UCI Machine Learning Repository. https://archive.ics.uci.edu/ml, 2020.

[Wäldchen *et al.*, 2021] Stephan Wäldchen, Jan MacDonald, Sascha Hauch, and Gitta Kutyniok. The computational complexity of understanding binary classifier decisions. *J. Artif. Intell. Res.*, 70:351–387, 2021.

[Wu *et al.*, 2023] Min Wu, Haoze Wu, and Clark W. Barrett. Verix: Towards verified explainability of deep neural networks. In *NeurIPS*, 2023.

[Yu *et al.*, 2023a] Jinqiang Yu, Alexey Ignatiev, and Peter J. Stuckey. From formal boosted tree explanations to interpretable rule sets. In *CP*, pages 38:1–38:21, 2023.

[Yu *et al.*, 2023b] Jinqiang Yu, Alexey Ignatiev, and Peter J. Stuckey. On formal feature attribution and its approximation. *CoRR*, abs/2307.03380, 2023.

[Yu *et al.*, 2023c] Jinqiang Yu, Alexey Ignatiev, Peter J. Stuckey, Nina Narodytska, and Joao Marques-Silva. Eliminating the impossible, whatever remains must be true: On extracting and applying background knowledge in the context of formal explanations. In *AAAI*, 2023.

[Yu *et al.*, 2024] Jinqiang Yu, Graham Farr, Alexey Ignatiev, and Peter J. Stuckey. Anytime approximate formal feature attribution. In *SAT*, pages 30:1–30:23, 2024.